The Effectiveness (or Lack Thereof) of Aim-Assist Techniques in First-Person Shooter Games

Rodrigo Vicencio-Moriera[1], Regan L. Mandryk[1], Carl Gutwin[1], Scott Bateman[2]

[1]University of Saskatchewan
Saskatoon, SK, S7N 5C9
firstname.lastname@usask.ca

[2]University of Prince Edward Island
Charlottetown, PEI, C1A 4P3
sbateman@upei.ca

ABSTRACT

Aim-assistance techniques have been shown to work for player balancing in 2D environments, but little information exists about how well these techniques will work in a 3D FPS game. We carried out three studies of the performance of five different aim assists in an Unreal-based game world. The assists worked well in a target-range scenario (study 1), but their performance was reduced when game elements were introduced in a walkthrough map (study 2). We systematically examined the relationships between realistic game elements and assist performance (study 3). These studies show that two techniques – bullet magnetism and area cursor – worked well in a wide variety of situations. Other techniques that worked well were too perceptible, and some previously-successful techniques did not work well in any game-like scenario. Our studies are the first to provide empirical evidence of the performance of aim assist techniques in 3D environments, and the first to identify the complexities in using these techniques in real FPS games.

AUTHOR KEYWORDS

Aim assistance; first-person shooter games; game balancing

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI).

INTRODUCTION

3D First Person Shooter (FPS) games are one of the most popular genres in computer gaming [10]. FPS games are complex and fast paced – this is especially true in multiplayer matches, where players need considerable skill to effectively navigate through a 3D environment, target opposing players, and avoid obstacles. FPS games take time to learn, resulting in a wide range of skill levels; experts are often vastly better at controlling the game than novices.

In many FPS scenarios, expertise is an important part of the experience: for example, some games have tournaments and rankings (e.g., Battlefield 3, Counter Strike, Call of Duty[1]). However, expertise differences also reduce the possibility of playing with others who have a different skill level; if skills differ greatly, the game is not fun for anyone [11].

Aiming is a critical component of FPS games, and aiming speed and accuracy is often a main differentiator between experts and novices. To improve novice targeting ability, *aim assistance* could be used. Aim assistance improves the accuracy and speed of target acquisition by manipulating factors such as the size of the target in motor space. Aim assistance techniques have been shown to improve targeting in 2D shooting games [4]. Aim assistance in these 2D situations led to closer matches between pairs and increased enjoyment, and the effects were not highly perceivable. Therefore, 3D versions of these aim-assistance techniques hold promise for assisting novices in FPS games.

Little information is available, however, about the use of these techniques in 3D FPS environments. Commercial games such as Halo appear to implement various forms of aim assist and previous work has shown that Fitts's law holds in 3D aiming tasks [19], so it is likely that applying aim assist techniques should help to improve performance in a 3D FPS. However, previous research has also shown that adding even simple game elements to a 2D aiming task can significantly change target acquisition time [14]. The complexity of a 3D FPS game environment (e.g., moving while aiming, moving targets, multiple targets, targets that shoot back, etc.) may interfere with the potential benefits of aim assist techniques, ultimately reducing their efficacy.

To investigate whether aim assist techniques are effective in 3D FPS games, we implemented five techniques (target lock, bullet magnetism, area cursor, sticky targets, and gravity) in a custom game environment and carried out three performance studies. In the first study (S1), we set up a simple 3D shooting range that was similar to the 2D games used in past evaluations of aim assistance. In the second study (S2), we created a realistic game level where players moved through a map with a number of computer-controlled players (bots). To begin teasing out the effects of realistic game elements, we carried out a third study that systematically varied two specific factors from S2: the effect of distractor targets, by removing friendlies from the map (S3A), and the effect of having more precise weapons, by switching to a semi-automatic sniper rifle (S3B).

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.
Copyright © 2014 ACM 978-1-4503-2473-1/14/04..$15.00.
http://dx.doi.org/10.1145/2556288.2557308

[1] battlefield3tournament.com, definitive-leagues.com, eswc.com

Our results showed that although all techniques worked well in S1, the presence of realistic game elements compromised the efficacy of most techniques – sticky targets and target gravity in particular. However, bullet magnetism and area cursor worked well in several settings, and were not highly perceptible. These two techniques are interesting in that they both change the targeting process *after* the user has completed their aiming motion and pressed the trigger; in contrast, the two worst techniques (gravity and sticky targets) both adjust the user's crosshairs during the aiming action. We speculate that there is value in letting the user plan and execute their own aiming motions, and only adjusting the shot during the flight of the bullet.

Our work is the first to explore the complexities of aim assistance in 3D environments, and the first to provide an analysis of real-game factors that can affect assistance. We also identify two successful candidate techniques that show potential for use in player balancing for FPS games.

RELATED WORK: GAME BALANCING

Flow describes a state where people are engaged and interested in their tasks without being overly anxious [8]. In order to provide players with a state of flow, game designers try to balance the level of challenge [11]. Striking the right balance is important, because a player's desire or willingness to play depends on whether the game is enjoyable [22, 24]. Unbalanced game elements may cause players to become frustrated or unengaged because they find the game too difficult or too easy [22]. For this reason, the game industry makes use of simulations and extensive play testing to discover unbalanced game elements [6].

To maintain flow in multi-player games, players' abilities to compete must be balanced. This means that players of equal skill should have an equal opportunity to win [1]. However, it can be difficult to balance the performance of different players [1] because people's abilities are based on many factors including their experience, reaction times, and skill [23]. Many games rewards skill, and newer or less-skilled players can have difficulty playing with more experienced players. Despite the difficulties in balancing competition in games, it has an important impact on player experience; a recent study demonstrated that balancing play of differently skilled players increases enjoyment for all [4].

Through our literature review and examination of games, we have identified four primary balancing approaches:

1. Difficulty Adjustment: Difficulty adjustment is one of the most common ways to allow users to adjust the level of challenge in a game. Many games allow players to explicitly select a difficulty setting, whereas others use dynamic difficulty adjustments, which change game elements based on in-game performance [17]; e.g., Left 4 Dead increases challenges if players do well. Dynamic adjustments in competition can lead to frustration if experienced players feel cheated [22], and can result in players feeling their achievements are meaningless [6].

2. Matchmaking: Many multiplayer online games use elaborate ranking systems that aim to group players with others of a similar skill level to ensure that the level of competition is balanced [22]. These work in a manner similar to ladders for sports like squash or tennis at sports clubs, and are important parts of Halo [16] and StarCraft 2.

3. Asymmetric Roles: In many games, players can take on different roles that suit their abilities or level of skill. For example, in Team Fortress 2, successful teams are composed of several different roles (e.g., spy, sniper, medic). This is analogous to real world sports teams, where players have very different roles (e.g., offense or defense).

4. Aim Assistance: Aim assistance facilitates the basic skills used in games that involve shooting, by making it easier to select on-screen targets. A recent study showed that aim assistance was an effective mechanism for balancing a simple 2D shooting-gallery game [4]. Players with less skill were able to improve and better compete with more skilled partners, and players reported having more fun when the competition was tighter. Further, participants did not notice that aim assistance was being given. However, there is little information about whether or not aim-assistance techniques would be effective in FPS games, which take place in a 3D environment and involve navigating an avatar through the space. Below we provide an overview on work in aim assistance and its use in video games.

RELATED WORK: AIM ASSISTANCE

Many aim assistance techniques have been devised to allow people to select on-screen targets quickly and accurately. Aim-assistance techniques are based on an understanding of aimed movements as modeled by Fitts's Law [13]. Fitts's Law states that the time to aim to a target is determined by the distance to reach the target (*amplitude*) and the size of the target (*width*). The smaller the amplitude and the larger the width, the easier and faster it is to point to a target. Aim-assistance techniques work in digital environments to facilitate pointing tasks using a device like a mouse or game pad by artificially manipulating amplitude and width [3]. Three general approaches for doing this are to reduce the distance to the object (amplitude manipulation), increase the width of the object in visual or motor space (width manipulation), or both [3]. There have been many aim-assistance techniques proposed (e.g., [4,5,7,9,12,15,18,20]). We refer the reader to a recent review (see [3]) for an overview, and we present individual techniques in the context of those implemented for the studies.

While work in aim assistance has been focused on its use in traditional 2D GUI pointing, a recent study has shown that Fitts's law also holds in a 3D FPS environment [19]. This suggests that aiming assistance techniques that have been developed in 2D should also work in 3D environments.

RELATED WORK: AIM ASSISTANCE IN GAMES

There are many genres where shooting (aim and select a target) is a core game element. Many are console games and use gamepads as input. Gamepads, where thumbsticks are used for aim, are not as precise as mice [21], so several aim-assistance techniques have emerged in FPS games that work similarly to techniques used in 2D GUI pointing. Our review identified three common approaches for aim assistance in games: bullet magnetism, target lock, and gravity. We discuss each of these in the next section.

TECHNIQUES CHOSEN FOR COMPARATIVE STUDIES

We chose to include five aim assist techniques in our work.

Target Lock

The Target Lock method (Lock) moves the crosshairs of the player to the closest target's head; Lock was activated when players pressed a button on the mouse (determined through pilot studies) The player's pitch and yaw were manipulated to make them face the closest opponent when the lock was activated. To vary the level of assistance, we varied the time to move the crosshairs to a target's head. At the lowest levels it was possible for an opponent to outrun being locked on to. At the highest level the crosshairs would be on the target virtually instantly. It took approximately 0.5s to lock on to a stationary target 90 degrees away at level 1 (roughly 7 degrees per tick) and 0.15s at level 10 (roughly 12 degrees per tick). In terms of Fitts's law, Lock reduces targeting time by reducing the distance between the pointer and the location of the target.

Target lock is not commonly used in FPS games but is popular in other genres. Games like Grand Theft Auto, Red Dead Redemption, and the 3D games in the Legend of Zelda series have used locking. This method is also based on the "object pointing" interaction technique, which tries to reduce empty space between targets by moving between selectable targets [15]. Lock may seem ideal for reducing targeting time but has drawbacks that may make it not viable in an FPS: distractor targets may compromise the performance of Lock; Lock requires explicit activation by the player; and Lock is obvious so players in competitive situations players may not enjoy it [2].

Bullet Magnetism

Bullet Magnetism (Bullet) "bends" bullets towards the closest target if a target is within the activation range (see Figure 1). The bullets in the game are instant shots, thus each bullet fired is described by a vector. When an enemy is within range (defined by 160 UDK units * Level, where $1 \leq \text{Level} \leq 10$), then Bullet adjusts this vector towards the closest opponent when the player fires and before the bullet collision logic. The higher the level of assistance, the farther away the effect begins and the more it corrects.

Bullet is not based on existing aim-assistance research but is present in games like Halo and Gears of War 3. It is exclusive to shooting games because of the presence of bullets. Bullet Magnetism adjusts the flight path of bullets once they leave the gun, and was implemented to mimic the way that bullet magnetism works in Halo. The closer the player is to a target when shooting, the more the bullet is attracted to the target. The end effect is similar to the area cursor method (described next) because it allows players to hit targets without perfect aiming, essentially increasing the target's width. We applied magnetism towards the body of the enemy if the crosshair was off the target and to the head of the enemy if the crosshair was already over a target. Assistance level was varied by the amount of magnetism applied. Bullet does not move the crosshair or change the CD ratio so may be less intrusive than other methods. However, Bullet may have issues if distractor target are present as it may be attracted to the wrong opponent.

Area Cursor

Area cursor (Area) follows the original implementation [18], but is adjusted to work in a 3D environment. Normally when a shot is made in the game a zero extent trace is used to determine if a target has been hit. When the area cursor method is active, a rectangle (non-zero extent) is projected into space and used to test intersection. On screen, the regular crosshair radius is 10px. With assistance, the physical size of the crosshair changes as the activation area changes, as seen in Figure 1. The size is calculated as 10px+(5px*Level), where $1 \leq \text{Level} \leq 10$.In terms of Fitts's law, area cursors essentially increase the width of the target [18], as such bigger cursors lower the index of difficulty. Assistance level is varied by varying activation area sizes.

Area cursor was chosen because it had the best performance results when compared with sticky targets or the gravity method in a 2D environment [4]. Additionally, Area has improved targeting for older adults [26] and users with motor impairments [12]. One advantage of Area is that it should still be effective with moving targets; however, sloppy targeting behavior may appear with players who get accustomed to the extra activation area [7], and distractor targets will affect Area. Our implementation handles multiple targets inside the activation area by choosing the target closest to the center of the crosshairs to be hit. Finally, we chose that our implementation would not directly help players get headshots; to score a headshot the center of the crosshair must be over the target's head.

Sticky Targets

The sticky targets method (Sticky) works by changing the control-to-display (CD) ratio when the crosshairs are over a target. The lower CD ratio results in a pseudohaptic effect of stickiness when over a target [7]. In our case, the amount of movement is divided by (Level + 4), where $1 \leq \text{Level} \leq 10$. In terms of Fitts's law, this method increases the width of the target in motor space [5] by affecting the second phase of pointing where corrective actions are taken. Once the player gets to the target and the stickiness kicks in they can position the cursor on the opponent more precisely due to the bigger size in motor space.

Sticky is used in games like Halo, Modern Warfare 2, and Red Dead Redemption (called reticule magnetism), and is generally applied on console games. We include Sticky because previous studies have shown that low levels of

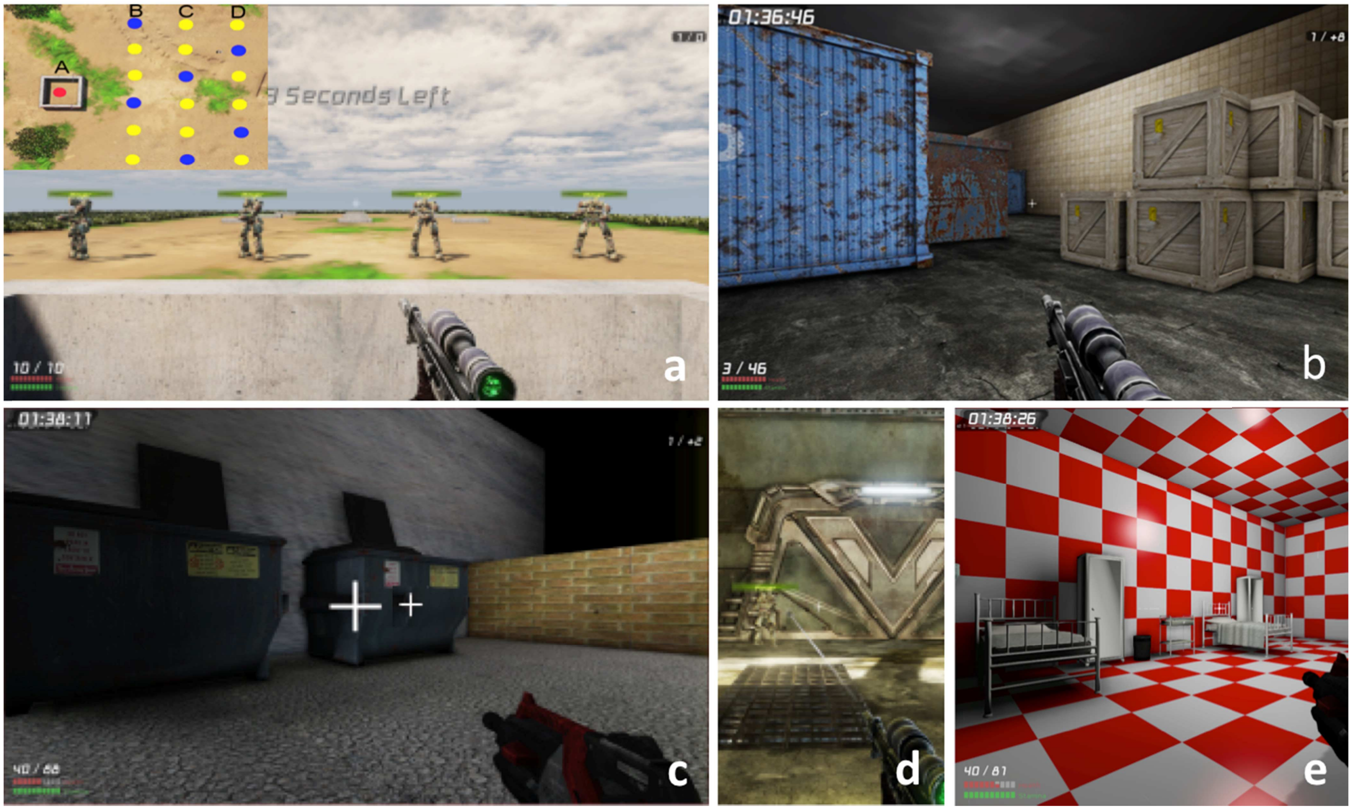

Figure 1. Images from the FPS environment. a) the player view of the shooting gallery; player and enemy locations are inset top-left; b) the realistic walkthrough; c) reticule showing low and high levels of Area Cursor; d) Bullet warping the hit location away from the crosshairs and toward enemy target; e)hospital room in the walkthrough.

stickiness are difficult to perceive [20]. Informal studies have also suggested that subjects are more accurate in a 3D FPS environment with CD adaptation than without [9]. However, distractors could be an issue with Sticky, compensation may lead to overshooting [20], and studies have suggested that this method might not be useful for moving targets [4]. Games like Call of Duty tend to move the cursor along with a target to mitigate the moving target problem; however, we did not implement this feature to better mirror the 2D version used in prior research.

Target Gravity

The target gravity method (Gravity) gives each target an attractive force that results in a player's crosshairs being dragged towards targets. In terms of Fitts's Law, Gravity reduces the distance to the target. There are several different ways of implementing target gravity; however, we transferred the 2D algorithms [4] directly into the 3D environment. The gravity is calculated by first identifying all targets within the player's line of sight. For these n targets, let $p_1, p_2,..p_n$ be the position of the targets, p_0 be the position of the player in the 3D space, and p_w be the warped position. G is the gravitational constant. For each target, the target weight is calculated with Equation 1. Then the warped position in 3D space is calculated with Equation 2.

$$w_i = \frac{G}{|p_0 - p_i|^2 + 1} \quad (1) \qquad p_w = \frac{\sum_{i=1}^{n} w_i p_i}{\sum_{i=1}^{n} w_i} \quad (2)$$

Equation 2 gives a position in the 3D environment that is the weighted average of all attractive forces. Because our 3D crosshairs do not have an actual position in the world (as opposed to the 2D implementation), we have to anchor the gravity effect. We calculate the current direction of the player and the direction of the warped position. The crosshair is then moved. A lower level of assistance moves the crosshairs more slowly than a higher level.

Gravity was included because of its strong performance in previous studies of 2D games [4]. Commercial games that use gravity, (e.g., console Call of Duty) try to only apply gravity effects on the closest target. In our implementation, we only activate gravity if the mouse is moved toward the warped position. If the mouse is not moved, or is moved away from the warped position, no effect is applied on the crosshairs. In addition, we implemented gravity to be attracted to the head of enemy targets to assist players with headshots. Gravity is subject to distractors, potentially moving the crosshairs when a player is aiming at another target.

STUDY DESCRIPTIONS

To test aim assist in FPS games, we conducted three studies.

Common Apparatus across Studies

We developed an FPS game using the Unreal Development Kit (UDK) to test the performance of the aim-assistance methods. The game was developed in UnrealScript, using

Visual Studio 2010 with the Nfringe add-on by PixelMine. The system ran on a Windows 7, Intel Core 2 Quad machine and an Acer 24-inch, 1920x1080 LCD monitor with a 60 Hertz refresh rate. Each participant set a comfortable sensitivity level for the Logitech G5 gaming mouse during the training round, and could not change it once the training rounds were complete. Logging of in-game actions was done on a Microsoft Sql Server 2008 R2 database. A C# console was written to receive logging messages from game clients and write them to the database.

Targets were shot by placing the center of the crosshairs on the target and left-clicking the mouse. The targets in S1 were stationary. In S2, the enemies moved in a random pattern when the player was in the enemy's line of sight. A random spot was chosen (within 300 UDK units from the current position) and the enemy began to move to that location while shooting at the player. Once the spot was reached, a new random location was chosen; this was repeated until the enemy was dispatched. The movement was limited so the enemies could not wander out of their rooms or move to the location of another enemy.

In both studies the player movement was controlled with a standard WASD scheme. Aim assistance levels were determined through pilot tests, and were custom-made. The walkthrough level was split into a hospital-themed area and a warehouse area. Both areas were linear so that the player could not explore, affecting the time measure. In the hospital area, enemies were placed in rooms along with friendlies. The warehouse area was darker and the enemies and friendlies were encountered as the player moved up. The locations of the enemies can be seen on the map in (Figure 2). The shooting gallery level was a simple outdoor environment. The player was confined to a square and the targets appeared in front of the player in several waves.

Participants were told that they would be testing several implementations of aiming assistance and that some rounds would have assistance, whereas some would not. Participants were unique to each study, which were all approved by the research ethics board.

Study 1 – Shooting Gallery

Study 1 (S1) extended previous work on modeling Fitts's Law in 3D FPS environments [19] and quantifying targeting assistance in 2D shooting games [4] with a shooting gallery-style task and 5 types of aim assist. User movements and enemy actions were constrained and the task focused on control rather than realism. The purpose of the study was to determine if the aiming assistance techniques that proved effective in 2D [4] transfer to 3D.

Task

Study 1 (S1) was a shooting-range-style level that was modeled after the level in the FPS Fitts's Law study by Looser et al. [19]. Participants used the sniper rifle (scope disabled), which took two body shots or one head shot to "kill" a target. The study consisted of one training round (no assistance), followed by the six technique rounds. In each round, there were seven waves of enemies. Each wave consisted of six targets; players tried to shoot as many as possible in 10 seconds. Waves 1 and 4 were close to the player; 2 and 5 were at a middle distance; 3 and 6 were farthest. In wave 7, enemies were placed in varying locations at the three distances, as shown in Figure 1.

In each of the six rounds, one type of aim assist was used: Bullet Magnetism, Area Cursor, Sticky Targets, Gravity, and Lock, all with the high level of assistance, and no assist (Control); a Latin Square balanced the presentation order.

Participants and Procedure

We recruited 12 players (11 male) to participate in S1. Three participants identified themselves as novices at FPS games, 5 as intermediate players, and 4 as experts. The experiment took 20 minutes to complete. After providing informed consent, participants completed the experimental task. Participants were compensated with $5.

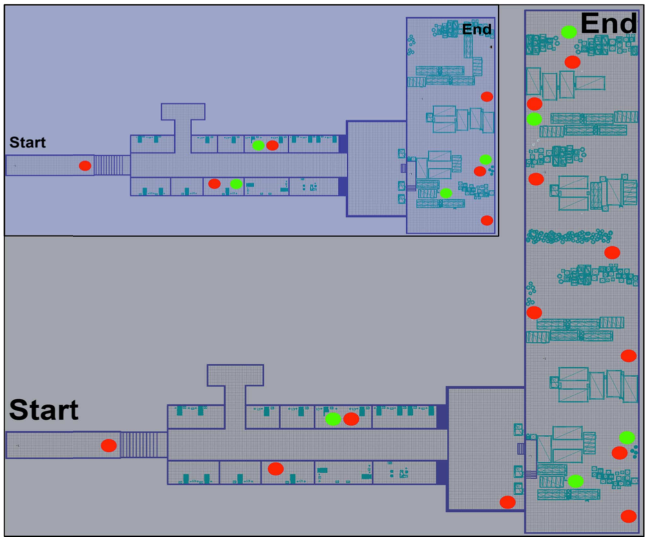

Figure 2. Walkthrough Maps: Grey Background–Full (S2); Inset Blue Background–Modified (S3). Red dots represent enemies; green dots represent friendly targets.

Study 2 – Full Walkthrough

Study 2 (S2) investigated the same 5 aim assist techniques in a realistic walkthrough task, complete with enemy targets who fired at the player and friendly targets that were to be avoided. Common features in FPS games (e.g., rifle scope) were implemented as we focused on game realism over control. The purpose of the study was to determine if the techniques that worked in the simple 3D environment would hold in a more complex "game-like" environment.

Task

S2 was a full game-like walkthrough level set in an abandoned warehouse (see Figure 2). The level contained 12 enemy bots and 8 friendly turtles that acted as distractor targets. Participants were equipped with an assault rifle and a pistol. The assault rifle had a higher rate of fire than the sniper rifle and did less damage with each hit. It took roughly 10 body shots or 5 head shots to kill a target with the assault rifle. The pistol had a slower rate of fire than the assault rifle, but did more damage per bullet (roughly five body shots or three head shots to kill a target). The level was complete with visual effects (e.g., lighting effects, glass shattering) to look and feel like a realistic FPS game.

Participants played the custom level 14 times. The first two rounds were used as training to familiarize players with the controls and locations of enemies and friendlies. The remaining 12 rounds consisted of one round with each of the six aim methods at low (Level 2) and high (Level 8) assistance and an additional two control rounds. At the end of each round, players completed a survey. Participants were instructed to avoid shooting the friendly turtles.

Participants and Procedure

We recruited 16 players (10 male) to participate in S2. Eight participants identified themselves as FPS novices and eight as experts. The same procedure as for S1 was used, except that due to the longer experiment time (between 80 min and 2 hours), participants were compensated with $15.

Study 3 – Full Walkthrough 2

Study 3 systematically varied two of the factors in Study 2 – presence of friendlies (S3A) and weapon type (S3B) – to quantify their relative effects on performance.

Task

Participants played two versions of the same map as S2 that was reduced in distance by half and contained six enemies (see Figure 2). The first version (S3A) used an assault rifle and had no friendly distractor targets. The second version (S3B) used a sniper rifle (scope disabled) and four friendly turtles. S3A and S3B omitted the Lock method and used only high (Level 8) assistance.

Participants went through the level 12 times. The first two rounds were training; the first was training for S3A and the second for S3B. The remaining ten rounds consisted of running through the control, bullet magnetism, area cursor, sticky targets, and gravity rounds in S3A and then repeating the process in S3B. The order of presentation of assistance types was balanced with a Latin Square. Half of the participants started with S3A and half started with S3B.

Participants and Procedure

We recruited 15 players (14 male) to participate in S3. Six participants identified themselves as FPS novices, five as intermediates, and four as experts. S3 used the same procedure as S2; participants were compensated with $10.

Dependent Measures

The same dependent measures were collected for each of the three studies, except there are no survey data for S1. **Accuracy:** *Hit Ratio* presents the number of hits over the number of shots fired, representing targeting efficiency. *Headshot Ratio* presents the number of headshots over the total number of hits, representing targeting quality. Subjective *Accuracy Rating* was provided by participants on a 21-pt scale (-10=least accurate to 10=most accurate). **Speed:** *Time* measured the elapsed time to complete the task. Subjective *Speed Rating* was provided by players on a 21-pt scale (-10=slowest to 10=fastest). **Outcome:** *Deaths* is the number of times a player was killed by an enemy. In S1, there were no enemies, thus *Kills* (the number of enemies killed) is used to reflect outcome (in S2 and S3, the player had to kill all enemies, thus there was no variance in kills). Subjective *Performance Rating* was provided by players on a 21-pt scale (-10=worst to 10=best). **Perception:** The *Perceptibility* of the assist technique was rated by players on an 11-pt scale (0=lowest, 10=highest).

Data Analyses

For each of the four data sets, we conducted RM-ANOVAs with Assist Technique as a within-subjects factor on all dependent measures. In S1 and S2 there were six levels of Aim Assist Technique (Control, Lock, Bullet Magnetism, Sticky Target, and Gravity); in S3A and S3B, there were five levels (no Lock). In S2, S3A, and S3A, we also included Expertise (novice or expert) as a between-subjects factor). Finally, S2 had an additional within-subject factor: level of Technique Strength (low or high).

Data violating the sphericity assumption had the degrees of freedom adjusted using the Huynh-Feldt method. Type 1 error was prevented by using the Bonferroni adjustment on all pairwise comparisons. α was set at 0.05 for all effects.

RESULTS

For all studies, statistical test results are shown in Table 1 and means and variances are shown in Figure 3. To improve readability, we summarize only main results here.

S1 – Shooting Gallery

Accuracy. All of the techniques improved the Hit Ratio over Control, except Gravity. Lock and Bullet also improved Headshot Ratio. **Outcome.** Lock, Bullet, and Area improved Kills over Control (and Gravity).

Summary of S1 Results

Gravity did not improve over Control on any measure. Sticky improved on Hit Ratio; Area improved on both Hit Ratio and Kills. Lock and Bullet improved on Hit Ratio, Headshot Ratio, and Kills. There were differences between the techniques as well, with Bullet and Lock beating the other techniques for Headshot Ratio and Kills.

S2 – Full Walkthrough

In S2, there were main effects of Technique Strength (level); the higher level resulted in a higher Headshot Ratio ($F_{1,14}$=14.9, p=.002, η^2=.52) and greater Perception ($F_{1,14}$=6.8, p=.021, η^2=.33). Because level produced few systematic differences, we present results for the higher level of aim assist only in S2 to correspond to the high level that was used in S1 and S3 (also in Table 1 and Figure 3).

Differences in Aim Assist Techniques

Accuracy. Lock and Area had higher Hit Ratios than Gravity. Lock and Bullet had higher Headshot Ratios than others. **Outcome.** Bullet and Lock resulted in fewer Deaths than Control. **Perception.** Lock was most noticeable.

Table 1. RM-ANOVA main effects for aim assist technique across all four data sets (columns) and measures (rows). Significant pairwise comparisons are shown (C=Control, B=Bullet Magnetism, A=Area Cursor, S=Sticky Targets, G=Gravity, L=Lock).

	Measure	S1 – Target Range	S2 – Realistic Level	S3A – No Friendlies	S3B – Sniper Rifle
Accuracy	Hit Ratio	$F_{5,55}$=23.3, p≈.000, η^2=.68 L, B, A, S > $C^{\ddagger}$, $G^{\ddagger}$	$F_{5,70}$=7.7, p≈.000, η^2=.35 L > C^{*}, G^{*} A > G^{*}	$F_{4,52}$=14.5, p≈.000, η^2=.53 B > $C^{\ddagger}$, $S^{\dagger}$, $G^{\dagger}$ A > $C^{\ddagger}$, S^{*}	$F_{4,52}$=9.4, p≈.000, η^2=.42 A > $C^{\ddagger}$, $G^{\ddagger}$
	Headshot Ratio	$F_{5,55}$=31.1, p≈.000, η^2=.74 L, B > $A^{\ddagger}$, $C^{\ddagger}$, $S^{\ddagger}$, $G^{\ddagger}$	$F_{5,70}$=53.5, p≈.000, η^2=.79 L, B > $A^{\ddagger}$, $C^{\ddagger}$, $S^{\ddagger}$, $G^{\ddagger}$ G > $A^{\dagger}$	$F_{4,52}$=60.0, p≈.000, η^2=.82 B > $A^{\ddagger}$, $C^{\ddagger}$, $S^{\ddagger}$, $G^{\ddagger}$ G > $A^{\dagger}$, C^{*}	$F_{4,52}$=64.6, p≈.000, η^2=.83 B > $A^{\ddagger}$, $C^{\ddagger}$, $S^{\ddagger}$, $G^{\ddagger}$ G > $S^{\dagger}$
	Accuracy Rating	N/A	$F_{5,70}$=7.8, p≈.000, η^2=.36 L > C^{*}, S^{*}, G^{*} B > G^{*}	$F_{4,52}$=3.27, p=.019, η^2=.20 No pairwise differences	$F_{4,52}$=4.8, p=.002, η^2=.27 B > $S^{\dagger}$
Speed	Time	$F_{5,55}$≈0.0, p≈1.00 (Controlled)	$F_{5,70}$=7.8, p≈.000, η^2=.36 L < C^{*}, G^{*}	$F_{4,52}$=9.48, p≈.000, η^2=.42 B < $C^{\dagger}$, $A^{\dagger}$, S^{*}, G^{*}	$F_{4,52}$=6.1, p≈.000, η^2=.32 B < C^{*} A < C^{*}, S^{*}, G^{*}
	Speed Rating	N/A	$F_{5,70}$=3.7, p=.007, η^2=.21 L > S^{*}	$F_{4,52}$=3.23, p=.019, η^2=.20 B > C^{*}	$F_{4,52}$=7.7, p≈.000, η^2=.37 B > A^{*}, $S^{\dagger}$, $G^{\ddagger}$
Performance	Deaths (Kills-S1)	$F_{5,55}$=84.7, p≈.000, η^2=.89 L, B > $A^{\ddagger}$, $C^{\ddagger}$, $S^{\ddagger}$, $G^{\ddagger}$ A > C^{*}, $G^{\ddagger}$; C > G^{*}	$F_{5,70}$=7.9, p≈.000, η^2=.36 B < $C^{\dagger}$, $S^{\dagger}$, G^{*} L < C^{*}	$F_{4,52}$=10.3, p≈.000, η^2=.44 B < $C^{\dagger}$, A^{*}, $S^{\ddagger}$, $G^{\dagger}$	$F_{4,52}$=8.7, p≈.000, η^2=.40 B < S^{*}, G^{*} A < $S^{\ddagger}$, G^{*}
	Performance Rating	N/A	$F_{5,70}$=5.3, p=.003, η^2=.28 L > $C^{\dagger}$, $S^{\dagger}$	$F_{4,52}$=4.6, p=.003, η^2=.26 B > C^{*}, $S^{\dagger}$	$F_{4,52}$=4.9, p=.002, η^2=.27 No pairwise differences
Perception	Perception Rating		$F_{5,70}$=8.1, p≈.000, η^2=.37 L > B^{*}, $A^{\dagger}$, $C^{\ddagger}$, $S^{\ddagger}$	$F_{4,52}$=4.8, p=.002, η^2=.27 $A^{\dagger}$, S^{*} > C	$F_{4,52}$=2.7, p=.041, η^2=.17 B > C^{*}

*p<.05, †p<.01, ‡p<.001

There were Expertise effects: novices took more Time ($F_{1,14}$=28.2, p≈.000, η^2=.67), had more Deaths ($F_{1,14}$=10.3, p=.006, η^2=.42), and a lower Hit ($F_{1,14}$=6.3, p=.025, η^2=.31) and Headshot Ratio ($F_{1,14}$=31.1, p≈.000, η^2=.69); however, there were no differences in technique based on expertise.

Summary of S2 Results

In general, players noticed the Lock method, and perceived that it aided their speed, accuracy, and overall performance. In terms of objective differences only Lock showed systematic improvements over Control in improved Time, Hit Ratio, and Headshot Accuracy, and fewer Deaths. Bullet magnetism also improved Headshot Ratio and Deaths over control, but there were no other improvements offered by the aim assist techniques over Control and differences between the assist techniques were minimal.

S3A – Assault Rifle, No Friendlies

Accuracy. Bullet and Area had higher Hit Ratios than Control or Sticky (Bullet higher than Gravity). Bullet and Gravity improved Headshot Ratio over Control and Area (Bullet higher than Sticky, Gravity). **Speed.** Bullet reduced Time over others. **Outcome.** Bullet reduced Deaths over others. **Perception.** Area and Sticky most perceptible.

There were no main effects of Expertise for any of the measures and no systematic differences between the aim assist techniques depending on expertise.

Summary of S3A Results

When friendlies were removed Bullet is the clear winner in terms of improved Hit Ratio, Headshot Ratio, Time, and Deaths. Interestingly, players did not notice Bullet, making it both effective and subtle. Other techniques also improved aspects: Area improved Hit Ratio but was perceived; Gravity improved Headshots and wasn't perceived; Sticky had no effect on performance, but was perceived.

S3B – Sniper Rifle

Accuracy. Area increased Hit Ratio over Control and Gravity. Bullet increased Headshot Ratio over all others. **Speed.** Area reduced Time over Control, Sticky, and Gravity. Bullet reduced Time over Control. **Outcome.** Bullet and Area reduced Deaths over Sticky and Gravity. **Perception.** Area and Sticky were more perceptible.

There were no main effects of Expertise for any of the measures; however, an interaction of Expertise and Aim Assist on Time ($F_{4,52}$=3.0, p=.027, η^2=.19) shows that experts took significantly less time than novices in the Bullet (p=.019) and Area (p=.001) conditions.

Summary of S3B Results

When the sniper rifle was used the techniques improved aiming differently. Area improved the Hit Ratio, whereas Bullet improved the headshot ratio. Both Bullet and Area reduced Deaths and the Time taken over the other approaches. In terms of noticeability, Area was perceived, whereas Bullet was not. Gravity had an elevated Headshot Ratio but no other differences, and Sticky resulted in no improvements, but was perceived.

Comparison of Techniques across Studies

The techniques differentially improved performance over Control under the various experimental conditions. **Gravity** underperformed across the board, and was the only approach that did not improve hit ratio in S1. **Sticky** helped with Hit Ratio in S2 but did not improve performance on any measure in the walkthrough tasks. Sticky only helps with targeting when the avatar and the enemies are both

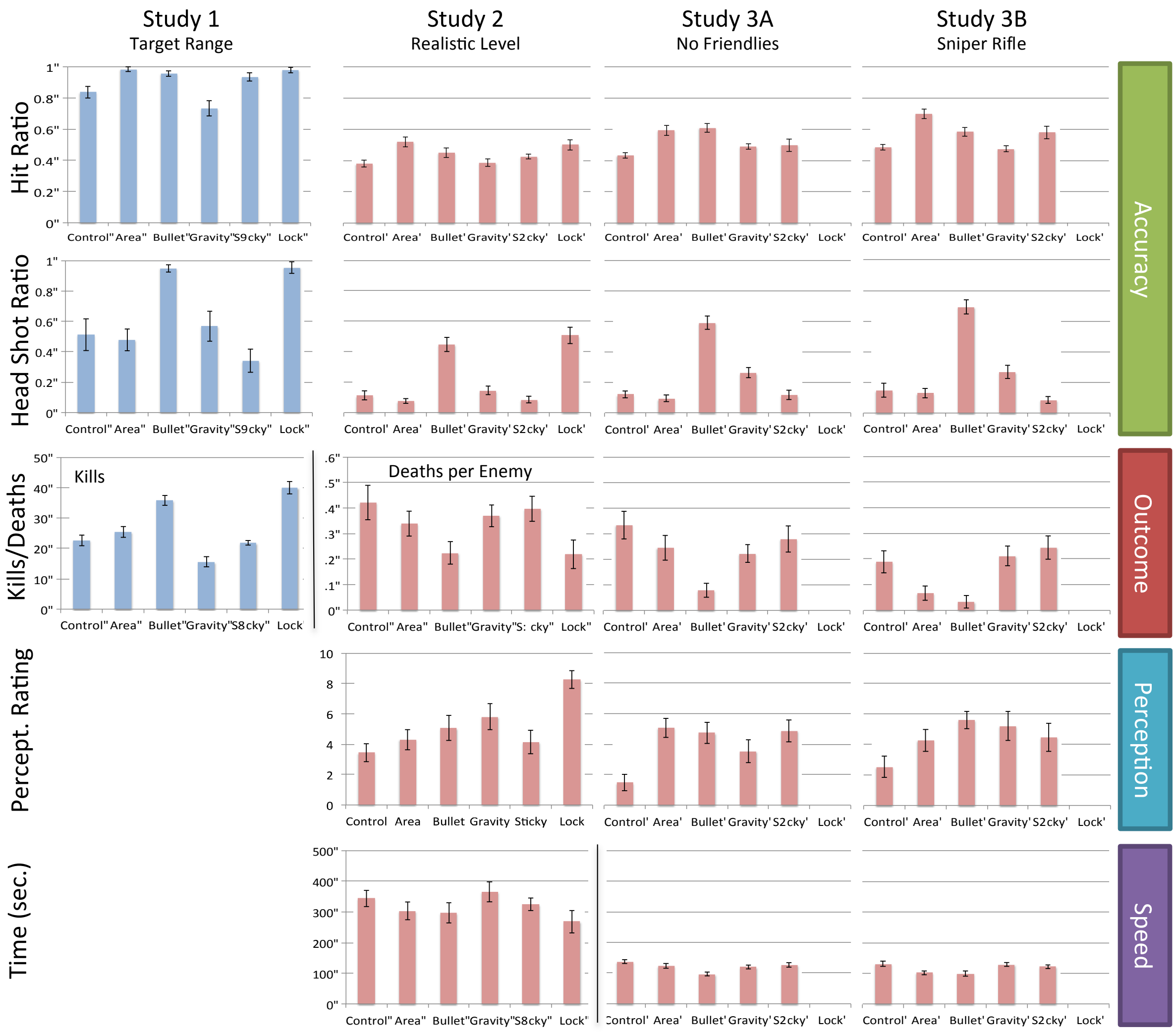

Figure 3. Means (±SE) from Studies 1 - 3. Vertical separators are used where comparison between studies is not possible. Line charts are used for a categorical x-axis to clearly show trends.

standing still. **Area cursor** improved Hit Ratio in S1, S3A, and S3B, but not S2. Area also improved kills/deaths in S1 and S3B, but did not work as well when the assault rifle was primarily in play (S2, S3A). **Bullet magnetism** worked well on all measures in S1, S3A, and S3B, but only worked for Headshot Ratio and Deaths in S2. **Lock** was only used in S2, and improved over Control on all objective measures.

DISCUSSION

The main findings from our studies are:

- Realistic game elements such as moving and aggressive targets, distractor friendlies, multiple-shot weapons, and lighting effects dramatically altered which aim assist techniques worked well in the 3D FPS;
- In our least-constrained game environment (S2), only target lock consistently outperformed the control condition (but was highly perceptible); bullet magnetism worked well for deaths and headshots;
- In the single-issue manipulations of study 3, bullet magnetism and area cursor performed well; gravity and sticky targets performed poorly;

Why did bullet magnetism and area cursor work well?

The bullet magnetism technique attracts a fired bullet, by a certain amount, to a point on the nearest target's head (if shooting at them) or body (if not shooting at them). Using the head as an attraction point is a clear explanation for this technique's success on the headshot measure; even when targeting the body, bullet magnetism could still bend the bullet's trajectory upwards to the enemy's head. Bullet magnetism also worked well both with a higher-precision weapon and with fewer distractor targets. These results are likely due to the technique's limited manipulation of trajectory: if the player can get the targeting reticule close to the intended enemy then the assist will improve targeting performance without causing errors.

Area cursor also performed well for several measures; this technique essentially shoots a larger bullet, increasing the target size. This effect worked particularly well with the higher-precision rifle – because there is no alteration to the aiming action, higher precision appeared to allow players to make faster use of the larger targets. Area cursor did not do well on the headshots measure, likely because of our implementation – unlike Bullet, if a user targeted the body with Area, the shot would hit the body, even if the area cursor's rectangle intersected the enemy's head.

It is interesting that both bullet magnetism and area cursor alter the targeting process only *after* the player has carried out their aiming action. These techniques do not change the user's control over moving their view, and do not attempt to 'decide' which target the user is aiming for. It is possible that magnetism and area cursor worked well because they caused less conflict with the user's control actions.

Why did gravity and sticky targets perform poorly?

Gravity and Sticky performed badly on almost all measures in all studies. Observations and player comments suggest that these techniques had problems because they change the movement of the reticule *while* a player is aiming, and this may cause conflicts with the user's intentions.

The gravity technique could drag the player's crosshairs to the wrong target (e.g., a friendly, or a different enemy than the user intended), and this would result in the user fighting against the assist. The gravity algorithm is attracted to the nearest target, but this may sometimes be at odds with where a user is trying to aim. This behavior may be specific to FPS games where a player plans to hit targets in a certain order – in contrast, in the 2D shooting gallery game where gravity worked well, targets were equally desirable and it mattered little if the technique hit the wrong target; the user could simply target the desired target on the next shot.

Sticky targets encountered similar problems – player ability to control an aiming movement was hampered by the assist because it manipulates the 'user movement phase' of targeting rather than the 'bullet travel' phase.

How can the effectiveness of techniques be judged?

In previous work in aim assistance for 2D pointing, and in applying aim assistance in a 2D game [4], hit ratio (i.e., a reduced error rate) was a primary metric for assessing success. This was the approach we initially used in judging our techniques in S1. However, with S2, it became clear that Bullet and Lock were the most effective techniques – participants died fewer times while completing the level, even though the hit ratio of Bullet was worse than other techniques that performed poorly in terms of outcome. Furthermore, Bullet was effective at increasing the number of headshots; because headshots are so important in Unreal, scoring a few headshots has more impact than scoring many body shots. When considering assistance techniques, designers should not focus only on hit ratio, but also be aware of key game mechanics, such as the importance of where hits register. While we expected that hit ratio would be the most important metric, it turned out to be only part of the larger picture when evaluating a complex game.

Practical Significance and Limitations

Our study looks at only how players respond when first using aim assistance (i.e., the early effects) and does not address what might happen after long-term repeated use (i.e., the long term effects). However, we feel that these early effects are more important to understand because the goal is for the techniques to be applied in a multiplayer environment to help weaker players compete. Balancing play actually *requires* early effects – the techniques have to work well for people who have not used them before. In addition, the weaker player will in all likelihood have less experience, and thus is it critical that the techniques work in unfamiliar environments with imperfect handling of the controls; realistically, novice players cannot anticipate enemies and may not be familiar with a certain game map.

The long-term effects are also important, which even our 2-hour study cannot assess. However, we have some guidance because half of our participants were FPS experts (and thus very familiar with the type of control and environment); the other half were novices. There were no systematic differences in the efficacy of the techniques depending on expertise. This is important because regardless of whether a novice is playing against an expert or an expert is playing against a professional, certain techniques will be more effective. So although we cannot predict the effects of long-term usage of the techniques themselves, we can assume that long-term usage of the general controls and environment do not affect technique efficacy.

The application of our results is limited to aiming in 3D games – our work is the first to investigate aim assistance in this realistic scenario, and generalizing to all 3D aiming is premature. However, this is not a niche topic – these games have huge player numbers and represent a massive industry that historically has not been guided by HCI research.

There are also aspects of our work that may apply beyond 3D FPS. The trend that post-aim techniques work better might be interesting to designers of other realistic and time-constrained systems (e.g., safety-critical or medical), whereas the controlled nature of S1 means that our results are of interest to others who study 3D aiming (e.g., VR).

Future Work

Our future work in this area will follow three directions. First, we will explore additional game factors– such as player movement, cost functions associated with shooting (e.g., in situations of limited ammunition), or additional precision (e.g., with a scope sight), and the role of lighting. Second, we will investigate the use of bullet magnetism and area cursor for balancing multiplayer games, using the modified Unreal system that we built for these studies. Third, we will further consider the way that the 3D environment itself affects assistance techniques, in terms of the user's movements in the X-Y plane versus the bullet's movement in all three dimensions – a characterization that has not been applied before to aim assistance techniques.

CONCLUSION

Aim-assistance techniques have been shown to work for player balancing in 2D studies, but little information exists about how well these techniques could work in a 3D FPS game. We carried out three studies to examine the performance of five different aim assists. Although the assists worked well in a target-range scenario, their performance was reduced when real-game elements were introduced. We examined the relationships between game elements and assist performance by manipulating specific game elements; we found that two techniques – bullet magnetism and area cursor – worked well in a wide variety of situations. In addition, we found that some techniques that worked well (i.e., target lock) were too perceptible to be successful for balancing, and that some previously-successful techniques (gravity and sticky targets) worked poorly throughout. Our studies provide initial insight into how game factors affect aim assistance, and identify techniques that show potential for player balancing.

ACKNOWLEDGMENTS

We thank our CHI reviewers, Jared Cechanowicz, the Interaction Lab, and NSERC and GRAND for funding.

REFERENCES

1. Adams, E. Fundamentals of Game Design. New Riders, 2010.
2. Adams, E. Difficulty Modes and Dynamic Difficulty Adjustment. *Gamasutra: The Art & Business of Making Games Blog*. Published May 2008, gamasutra.com/ view/feature/3660/the_designers_notebook_.php
3. Balakrishnan, R. “Beating” Fitt‟s law: virtual enhancements for pointing facilitation. IJHCS, 61, (2004), 857-874.
4. Bateman, S., Mandryk, R.L., Stach, T. and Gutwin, C. Target assistance for subtly balancing competitive play. In CHI 2011, 2355-2364.
5. Blanch, R., Guiard Y., and Beaudouin-Lafon, M. Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation. In CHI 2004, 24-29.
6. Bostan, B., and Öğüt, S. Game challenges and difficulty levels: lessons learned From RPGs. In International Simulation and Gaming Association Conference, 2009.
7. Cockburn, A, and Firth, A. Improving the Acquisition of Small Targets. In HCI03 Conference on People and Computers XVII, 2003. 181-196.
8. Csíkszentmihályi, M. Flow: The Psychology of Optimal Experience. Harper and Row, 1990.
9. Elmqvist, N., and Fekete, J. Semantic Pointing for Object Picking in Complex 3D Environments. In GI 2008, 243-250.
10. Entertainment Software Association. *2013 Essential Facts About the Computer and Video Game Industry* www.theesa.com/facts/pdfs/ESA_EF_2013.pdf
11. Falstein, N. Understanding fun – the Theory of Natural Funativity. In Introduction to Game Development, Rabin, S. (ed.), 71-98. Charles River Media, 2005.
12. Findlater, L, Jansen, A., Shinohara, K., Dixon, M., Kamb, P., Rakita. J., and Wobbrock, J.O. Enhanced Area Cursors Reducing Fine Pointing Demands for People with Motor Impairments. In UIST 2010, 153-162.
13. Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. J.Exp. Psych., 47, 381-391.
14. Flatla, D., Gutwin, C., Nacke, L.E., Bateman, S., and Mandryk, R.L. Calibration Games: Making Calibration Tasks Enjoyable by Adding Motivating Game Elements. In UIST 2011, 403-412.
15. Guiard, Y., Blanch, R., and Beaudouin-Lafon, M. Object pointing: a complement to bitmap pointing in GUIs. In GI 2004, 9-16.
16. Herbrich, R., Minka, T., and Graepel, T. TrueSkill(TM): A Bayesian skill rating system. In Advances in Neural Information Processing Systems, 20, 569-576, 2007.
17. Hunicke, R. The case for dynamic difficulty adjustment in games. In ACE 2005, vol. 265, 429-433.
18. Kabbash, P. and Buxton, W. The "Prince" Technique: Fitts’ Law and Selection Using Area Cursors. In CHI 1995, 273-279.
19. Looser, J., Cockburn, A. and Savage, J. On the Validity of Using First-Person Shooters for Fitts' Law Studies. In HCI 2005, 33-36.
20. Mandryk, R.L., Gutwin, C. Perceptibility and Utility of Sticky Targets. In GI 2008, 65-72.
21. Natapov, D., Castellucci, S. J., and MacKenzie, I. S. (2009). ISO 9241-9 evaluation of video game controllers. In GI 2009. 223-230.
22. Newheiser, Mark. Playing Fair: A Look at Competition in Gaming. *Strange Horizons*, www.strangehorizons. com/2009/20090309/newheiser-a.shtml
23. Sweetser, P. and Wyeth, P. GameFlow: a model for evaluating player enjoyment in games. Computers in Entertainment, 3, 3-3, 205.
24. Vorderer, P., Hartmann, T., and Klimmt, C. Explaining the enjoyment of playing video games: the role of competition. In Entertainment Computing’03, 1-9.
25. Wobbrock, J.O., Fogarty, J., Liu, S., Kimuro, S., Harada, S. The Angle Mouse: Target-Agnostic Dynamic Gain Adjustment Based on Angular Deviation. In CHI 2009, 1401-1410.
26. Worden A, Walker N, Bharat K, Hudson S. Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons. In CHI 1997, 266-271
27. Youssef, A. and Cossell, S. Thoughts on adjusting perceived difficulty in games. In Australasian Conference on Interactive Entertainment, 1-4, 2009.